To create the hangman solution in GO, I stored the users’ input of their level choice and used switch statements to handle the different cases. I used the rand package and time package in GO to create a random number generator that takes in the current time so that each time the rand.New(rand.NewSource(time.Now().UnixNano())) line is used, a different number is generated. This leads to a different index being used and therefore a different word. I also used time handling to measure the time passed during the game and to implement a time limit for each level to increase the difficulty.

The chosen paradigm for the GO code was functional. The code has functions that hold repeated blocks of code. The solution has playHangman function which hold the most code of the hangman game and is called within the main() body of the program. The code has multiple functions for example chooseWord, when it is called it chooses a random word from the word arrays and returns the word to where it was called from. The code implements immutability when using the guessedLetters array as it appends a new guessed letter to the guessedLetters slice rather than modifying a previous one.